# Computation with Unitaries and One Pure Qubit

D. J. Shepherd*
*University of Bristol, Department of Computer Science*

November 28, 2006

**Abstract**

We define a semantic complexity class based on the model of quantum computing with just one pure qubit (as introduced by Knill & Laflamme) and discuss its computational power in terms of the problem of estimating the trace of a large unitary matrix. We show that this problem is complete for the complexity class, and derive some further fundamental features of the class. We conclude with a discussion of some associated open conjectures and new oracle separations between classes.

## 1 Introduction

The idea of doing quantum computation with just one pure qubit is relevant to certain problems in the theory of quantum chaos [5], and possibly relevant in terms of choosing an architecture to pioneer practical quantum computation, since at present there would seem to be little agreement about which technology is most fit for implementing a general purpose quantum processor, [1]. Yet the literature seems to lack a discussion of the complexity classes naturally associated to the one-pure-qubit paradigm. Taking a computer science perspective, we will define a class analogous to **BQP** using the circuit model, discuss some basic conjectures about that class, provide a definition of relativisation for the model, and show some evidence for the conjectures in terms of that relativisation. It is hoped that this will help promote the search for algorithms in other paradigms.

DQC1 as a computational model, ("Deterministic[1] Quantum Computation – 1 pure qubit", as defined in [5]) is an apparently

*shepherd@compsci.bristol.ac.uk, dan.shepherd@cesg.gsi.gov.uk
[1]The adjective "Deterministic" will not prove to be especially relevant throughout.

less powerful computational idea than 'full' quantum computation, yet it appears to be able to do certain things exponentially more quickly than can a classical device. In particular, the original paper [5] explains something of the relation to the classically hard problem of trace estimation. In this article, we use the idea of computational circuits to derive a proper complexity class for the DQC1 model, in order to ask about the languages that can be resolved with bounded probability in polynomial time, and we address conjectures concerning the ability of families of such circuits to resolve some of the 'classical' languages belonging to classes such as **P** and **BPP**.

There is a nice description of general quantum circuits in [4], where it is shown how circuits can naturally be composed whenever the output of one circuit is of the same "type or kind" as the input to the next. The authors also point out that it can sometimes be useful to have the elements of a quantum circuit depend on a classical string that has the status of 'input', to allow for notions of adaptation. Both of these themes are key to the development of a computational class that expresses the power of the DQC1 model.

This first section recaps some basic notation used ubiquitously within Quantum Information Science, recaps some definitions of *circuits* as used in context of computational complexity theory, and introduces a new complexity class for capturing the power of the DQC1 methodology within the context in which we're interested. Section 2 illustrates in detail a well-known complete problem for the "one pure qubit" approach, and shows how our complexity class is relevant to that problem. Section 3 looks in more detail at the new class, proving some basic properties about it and showing that it is closed under certain kinds of reduction. Section 4 makes some conjectures about how our complexity class stands in relation to other common classes, analysing the strength of the conjectures with regard to known hard challenges of complexity theory, and providing 'evidence' for believing the conjectures despite an absence of formal proof.

## Algebra Notation

We use the algebra formally defined by

$$\mathcal{A}_w \;:=\; \frac{\mathbb{C}\big[\; X_i, Z_i \;\big]_{i=1}^{w}}{\big\langle\; X_i^2 - 1, Z_i^2 - 1, (X_i, Z_i), [X_i, X_j], [X_i, Z_j] \;\big\rangle_{i \neq j}}, \quad (1)$$

where $w$ is a positive integer, the *width* (in qubits) of a quantum circuit. This algebra is nothing other than the ring of square ($2^w$-

by-$2^w$) complex matrices, but it is useful to be able to refer to the algebra without having to speak explicitly of matrices. On it we can define Trace ($Tr$) and adjoint ($\dagger$) in the usual manner, these having the properties that one would expect of a matrix algebra. (Note that the value of $Tr[1]$ will depend on which $\mathcal{A}_w$ the 1 is taken from.) $X$ and $Z$ denote the usual Pauli operators. The unitary transformations of quantum physics correspond with the automorphisms $\rho \mapsto U \cdot \rho \cdot U^\dagger$, for unitary $U$.

The DQC1 model is characterised by requiring the starting state of any computation to be highly mixed except for one pure qubit : the starting state is given as the (Hermitian) density operator

$$\rho_{\,\mathrm{start}} \;:=\; \frac{1+Z_1}{2^w} \;\in\; \mathcal{A}_w. \tag{2}$$

The formalism must expressly forbid any non-unitary gates, since the use of 'zeroise' gates, for example, will readily enable one to purify more qubits and then perform a computation on many pure qubits. Instead, the *only* allowed actions on the density operator (within DQC1,) besides a final measurement, are conjugations by unitary operators (*i.e.* inner automorphisms.) Thus the states (density operators) reachable from the starting state using only these allowed automorphisms have the form

$$\rho \;=\; \frac{1+UZ_1U^\dagger}{2^w} \;=:\; \frac{1+\beta Z_1+\sqrt{1-\beta^2}R}{2^w}, \tag{3}$$

where $R$ and $Z_1R$ are traceless, and $\beta$ is real. This format for writing a generic state is chosen so as to highlight the coefficient in front of the $Z_1$ term, which we will later define as the 'output' of a circuit whose gates implelemt the unitary $U$. That $\beta$ is real can be seen from its definition as $\beta := Tr(UZ_1U^\dagger Z_1)/2^w$ and the fact that $UZ_1U^\dagger Z_1$ is a product of two hermitian operators.[2] That $R$ and $Z_1R$ are traceless can be seen from the definition $R := \frac{UZ_1U^\dagger-\beta Z_1}{\sqrt{1-\beta^2}}$. (We leave $R$ undefined in the cases where $\beta^2 = 1$.)

We think of $R$ as coding the state of the 'workspace' of a DQC1 algorithmic process, and $\beta$ as coding the 'output' of the process. The final measurement, to be applied to the state after the completion of the unitary circuit, will measure the first qubit in the computational basis. The probability of obtaining "0" from this reading for the state described in equation (3) is

[2] $(AB)^\dagger = B^\dagger A^\dagger = BA; \quad Tr[BA] = Tr[AB]$.

therefore given by

$$\mathbb{P}(\text{“0”}) \;=\; Tr\left[\rho \cdot \frac{1+Z_1}{2}\right] \;=\; \frac{1+\beta}{2}. \tag{4}$$

## Entanglement

It is worth mentioning here something about the entanglement of states of the form of equation (3). While we won't be considering any particular applications of entanglement, we note that entropy considerations alone guarantee that, for any reasonable measure, there won't be *much* entanglement present in such a system. But it would be inaccurate to say that such states are necessarily free of *all* quantum correlations. An example of an entangled state would be

$$\rho_{\text{Ent}} \;:=\; \frac{1}{2^{w+1}}(\; 2 + X_1X_2 - Y_1Y_2 + Z_1 - Z_2 \;), \tag{5}$$

which is prepared from the state $\rho_{\,\text{start}}$ by the unitary operator

$$U \;=\; \frac{1}{2}(\; 1 + Z_1 + X_2 - Z_1X_2 \;) \cdot \frac{1}{2}(\; 1 - Z_2 + H_1 + H_1Z_2 \;). \tag{6}$$

To see that it is entangled, note that $\rho_{\text{Ent}}$ satisfies

$$Tr[\; \rho \cdot (1 - Z_1)(1 + Z_2) \;] \;=\; 0, \tag{7}$$

$$Tr[\; \rho \cdot (X_1 + iY_1)(X_2 + iY_2) \;] \;\neq\; 0. \tag{8}$$

However, any pure product state

$$\frac{1}{2^w}(1 + \alpha_1X_1 + \beta_1Y_1 + \gamma_1Z_1) \cdot (1 + \alpha_2X_2 + \beta_2Y_2 + \gamma_2Z_2) \cdots \tag{9}$$

either satisfies equality (7) and then fails the inequality (8), or else satisfies the inequality (8) but then fails the equality (7) by contributing a strictly positive (real) amount. Therefore $\rho_{\text{Ent}}$ cannot be a convex combination of pure product states.

## Circuits in general

Here we recap a few basic definitions for circuits and computation via circuits. A *classical* deterministic circuit is usually defined (*e.g.* [2]) as a directed acyclic graph whose vertices are either 'inputs' or 'constants' (with no indegree), 'NOT gates' (with indegree 1), or 'AND' or 'OR' gates (with indegree 2). The vertices with no outdegree are called 'outputs'. The language obtained

from an infinite family $\{C_1, C_2, ...\}$ of such circuits, where circuit $C_n$ has $n$ inputs and one output vertex, is the set of strings

$$\mathcal{L} \;=\; \{\; x \in \{0,1\}^* \;:\; C_{|x|}(x) = 1 \;\}. \tag{10}$$

Here $C_n(x)$ denotes the output of circuit $C_n$ when the gates are evaluated after input string $x$ (of length $n$) is used to load boolean values into the 'input' vertices. Then whenever the family is described by a logspace Turing machine, we have that $\mathcal{L} \in \mathbf{P}$. Such families are called *uniform*, and are always polynomially bounded [2].

There are many ways to introduce randomness to this model. For example, one may allow for extra vertices of zero indegree that are to be initialised randomly, or one may incorporate 'coin-flip' vertices with indegree 1 that 'overwrite' data with random data. Within models which allow for some randomness, we redefine $C_n(x)$ to be the *probability* of the output vertex evaluating to "0". If there's a guarantee that $C_{|x|}(x) \notin (\frac{1}{3}, \frac{2}{3})$, for all strings $x$, then a language in **BPP** may be derived. This class is a semantic class since it depends on a guarantee. There is nothing special about the $(\frac{1}{3}, \frac{2}{3})$ limitation in that guarantee, since any non-negligible separation in the probabilities can be amplified by repeated computation [2].

Quantum circuits are usually defined similarly [1] with unitary gates replacing ordinary vertices, and measurement gates relpacing output vertices. Unitary gates have the same indegree as outdegree, because they are reversible. Measurements are usually taken to be single-qubit measurements in the computational basis, (see [1] for more details.) Thus is the class **BQP** derived. As well as being aptly notated pictorially by a directed acyclic graph, a quantum circuit can equally well be notated simply as a sequence of the unitary operators applied, which is to say we can describe it by listing a series of unitary elements from $\mathcal{A}_w$, to be applied sequentially to some starting input. Without loss of generality, all measurement can be delayed to the end of the quantum circuit, and since only one bit of output is required for decision problems, that output measurement can then be a measurement of $Z$ on the first qubit, with other qubits being traced out.

## DQC1 Circuits

For DQC1 circuits we approach the problem slightly differently, because we need a different notion of 'circuit input' to satisfy the DQC1 framework of allowable states. Because the actual

quantum input into the circuit must necessarily be of the form prescribed by equation (2) for DQC1 circuits, we need an alternative way of getting the input string $x$ into the computation. So we recycle the idea of a circuit being a list of unitary operators from $\mathcal{A}_w$, but each element of the list will be a *choice of two* unitary operators, the actual one to apply being selected by a (classical) bit of $x$. Thus an example of a description of a small DQC1 circuit would be $(A\backslash B)_{x_1} \cdot (C\backslash D)_{x_2} \cdot (E\backslash F)_{x_1}$. This circuit, on input $x =$ "01", would perform the automorphism specified by conjugation by the unitary $A \cdot D \cdot E$ on the starting state, but on input $x =$"11" would conjugate the starting state by $B \cdot D \cdot F$, *&c.* In general, we allow a polynomially bounded circuit width, $w$, so $A, B, C, D, E, F$ must be unitary elements of $\mathcal{A}_w$. (Note that this definition of input could equally well have done for the classes described above, though it is not commonly employed.)

The circuit output probability $C_n(x)$ is then taken to be $(1+\beta)/2$ as per equations (3) and (4), where $\beta$ is found from the state resulting from the starting state (equation (2)) being subject to the specified sequence of automorphisms. As before, $C_n(x)$ has the physical meaning of being the probability of measuring the first qubit of the final state to be $|\,0\,\rangle$. (An example of a problem that fits this paradigm – indeed the canonical example – is that of estimating the trace of a unitary matrix. The input to the problem will amount to a classical description of the matrix. The details are described below in section 2.)

Thus is the class **BQ1P** derived[3], assuming as before that the family of circuits is uniform. The precise rule that we adopt for defining **BQ1P** languages interprets probabilities a little more generally too; instead of requiring that $C_{|x|}(x) \notin (\frac{1}{3}, \frac{2}{3})$ we simply require the (possibly) weaker condition that for $n = |x|$

$$C_n(x) \;\notin\; \left( \frac{1}{2} - \frac{1}{2q(n)},\; \frac{1}{2} + \frac{1}{2q(n)} \right), \tag{11}$$

for some known polynomially bounded complexity function $q(n)$. The reason for allowing polynomially small error bounds is that there is no apparent way to 'chain together' (sequentially compose) or even *parallel compose* DQC1 circuits, and so unlike the **BPP** or **BQP** cases, amplification of probability has to take place *outside* of the model. And so any language in **BQ1P** must

[3]This is our notation, "Bounded-error Quantum – 1 pure qubit – Polynomial time computation", but we welcome correction from those who know better how to name new classes.

satisfy

$$\mathcal{L} := \left\{ x \in \{0,1\}^* \ : \ \beta(x) \ \geq \ \frac{1}{q(|x|)} \right\}, \qquad (12)$$

$$\bar{\mathcal{L}} := \left\{ x \in \{0,1\}^* \ : \ \beta(x) \ \leq \ \frac{-1}{q(|x|)} \right\}.$$

The class is reasonably robust against variation in the exact specification of which unitaries are allowed within the circuit. Since the circuit families must be uniform (*i.e.* describable by a well-behaved logspace Turing machine,) it will be important to avoid encoding any 'complexity' in the alphabet of allowable unitaries. In general we will take a finite alphabet of gates and then work with its closure under conjugation by SWAP gates. The resulting alphabet (subset of $\mathcal{A}_w$) will be of polynomially bounded size, and we shall want for it to form a universal set in the sense of being able to generate any unitary to arbitrary precision as $w \to \infty$. An example of a such an alphabet would be the set of all CNot unitaries $\frac{1+Z_j+X_k-Z_jX_k}{2}$ together with the set of all Hadamards $\frac{X_j+Z_j}{\sqrt{2}}$ and all '$\pi/8$' unitaries $\exp(iZ_j\pi/8)$. Robustness of classes with respect to changes in the defining gate-set is discussed amply in the literature (*e.g.* [1].)

## 2 Completeness

Although **BQ1P** is a semantic class, we can still provide a notion of completeness for it, by reference to the problem of estimating the (sign of the real part of the) trace of a unitary operator. This problem is discussed in [5], and we recap and expand on the main ideas here.

Formally, the promise problem for Trace Estimation is : given a description of a unitary matrix in factored form (see below), decide whether the sign of the real part of the trace of the matrix is positive, given the promise that the magnitude of the real part of the trace is polynomially bounded away from zero. Specifically, this means that if $x$ is the description of the unitary $U$, and $U$ belongs to the algebra $\mathcal{A}_w$, then the promise holds that $\left| \ Re[Tr[U]] \ \right| \geq \frac{2^w}{p(|x|)}$ for some polynomially bounded complexity function $p$. The dependence on $w$ serves to make the problem well-defined, essentially remedying the fact that Trace applies a linear multiplier of $2^w$. It may be noted [5] that for a randomly chosen $U$ without any such promise, the typical value of $\left| \ Re[Tr[U]] \ \right|$ is $\sqrt{2^w}$, and that in any case the value is constrained to the region $[0, 2^w]$.

Let $(L_{w,k})_{w,k}$ be some sensibly chosen *uniform* family of maps that interpret binary strings of length $k$ into unitary elements of $\mathcal{A}_w$; (the details are unimportant.) Let $|x| = kt$ and let

$$U(x, w, k) \;=\; L_{w,k}(x_1 \dots x_k) \cdots L_{w,k}(x_{kt-k+1} \dots x_{kt}), \qquad (13)$$

denote the unitary matrix that is obtained by parsing the string $x$ in sections of $k$ bits, using $L_{w,k}$, and composing the resulting network of $t$ gates (each acting on $w$ qubits.)

**Lemma 1** *The promise problem for Trace Estimation is complete (using classical logspace reductions) for* **BQ1P**.

To show completeness of the promise version of the Trace Estimation problem for **BQ1P** we need to do two things : first we shall show that this version can be solved within the **BQ1P** methodology, and second we shall show that any other methodology solving the promise problem can simulate the action of any other **BQ1P** computation. This will not however help us write down any particular language in **BQ1P**, because the promise in question cannot be readily encoded in the usual linguistic syntax. The notion of reduction that we use in reducing other problems to this Trace Estimation problem will be reduction by classical processing bound to logarithmic space [2]; see lemma 4 in section 3 for more details.

Proof part 1) (*cf* [5]) For the first part of the proof, we need a uniform family of DQC1 circuits such that circuit number $(|x|, w, k)$, of width $w'$, on input $x, \rho_{\,\mathrm{start}}$, will generate a state for which $\beta(x) \geq 1/q(|x|)$ if and only if $Re[Tr[U(x, w, k)]] \geq 0$, *etc.* To achieve this, we take $w' = w + 1$ and use a DQC1 circuit that begins and ends with a Hadamard gate on the first qubit, and in between these performs the relevant conditional unitaries on the remaining qubits, controlled also on the (quantum) setting of the first qubit to the state $|\,1\,\rangle$. (The factors in equation (13) can be broken down in a uniform way[4] into choices of unitaries controlled by the classical bits of $x$, and for any fixed alphabet, it is a finite problem to break down controlled versions of gates into simpler forms allowed in the alphabet, with arbitrary accuracy.) Writing this out for any specific input $x$ gives

$$H_1 \cdot \Lambda_1(U(x, w, k)) \cdot H_1. \qquad (14)$$

This circuit, on input $\rho_{\,\mathrm{start}}$, will yield the $\mathcal{A}_{w+1}$ state

$$\frac{1 \;+\; \frac{U(x,w,k)+U^\dagger(x,w,k)}{2} \cdot Z_1 \;-\; i \cdot \frac{U(x,w,k)-U^\dagger(x,w,k)}{2} \cdot Y_1}{2^{w+1}}; \qquad (15)$$

[4] *cf* section 3

for which the $\beta$ of equation (4) is $Re[\, Tr[\, U(x,w,k) \in \mathcal{A}_w \,]/2^w \,]$. Invoking the promise, we can be sure that this $\beta$ satisfies $|\beta| \geq \frac{1}{p(|x|)}$, and so by taking $q > p$, we can ensure that the correct decision is made according to equation (12), even allowing room for small errors in any approximations that might be required when rendering equation (14).

Proof part 2) For the second part of the completeness proof, suppose we have some 'black box' means for deciding of the triple $(x,w,k)$ whether the real part of the trace of $U(x,w,k)$ is positive, given that it is bounded away from zero. We shall use this black box, together with what amounts to merely logspace processing, to decide an arbitrary **BQ1P** language. The use of logspace processing will be 'justified' later in section 3.

An arbitrary **BQ1P** language that we would have our black box decide may be written as

$$\left\{ (y,w,k) \;:\; Tr\left[ \frac{U(y,w,k) \cdot Z_1 \cdot U^\dagger(y,w,k) \cdot Z_1}{2^w} \right] \geq \frac{1}{q(|y|)} \right\}, \tag{16}$$

as derived from equations (3), (4), and (12). Then all we need do is to have $x$ code for a related circuit of the same width and similar depth, so that $U(x,w,k) = U(y,w,k) \cdot Z_1 \cdot U^\dagger(y,w,k) \cdot Z_1$ and $|x| = O(|y|)$; for then the black box will be able to decide on $(y,w,k)$ via $x$. This will be a simple matter for any reasonably chosen $L_{w,k}$. ■

# 3 Properties of DQC1 Circuits

This section gathers together some important properties of the circuits we've been discussing, and of **BQ1P** itself.

### Boolean formulas on inputs

Observe that the following small DQC1 circuit will 'compute' the boolean operation $x_1 \wedge x_2$, deterministically, using a circuit of total width $w \geq 1$ :

$$(1 \backslash H_1)_{x_1} \cdot (1 \backslash Z_1)_{x_2} \cdot (1 \backslash H_1)_{x_1}. \tag{17}$$

More precisely, this will map the starting state to $(1 + \beta Z_1)/2^w$ where $\beta = (-1)^{x_1 \wedge x_2}$. Likewise, other simple boolean gates may be rendered. Note that it is necessary to use one of the bits twice (bit $x_1$ in this example) in order to perform a non-trivial gate. In general, it would be exceptionally limiting if we were

only allowed to use each input bit once, because after each input bit had been used, no further automorphisms would be able to change any of the algebraic relations held between the states arising from different $x$ values.

Does the ability to perform these simple boolean gates mean that it is straightforward to put $\mathbf{P} \subseteq \mathbf{BQ1P}$? Well, no, because there is no method prescribed to take the output of one DQC1 circuit directly into the input of another DQC1 circuit, and therefore it is not enough to be able to perform interesting gates only on *input bits*, one must also be able to perform interesting gates on *intermediate bits* (whatever they might be.) Recall from equation (3) that there is a somewhat limited 'place' in which 'intermediate data' can be held during a DQC1 computation.

### Complement

It is obvious from the definitions that $\mathbf{co\text{-}BQ1P} = \mathbf{BQ1P}$, and so we shall say no more about this, except to point out that the sign of $\beta$ in equation (3) can be flipped at any time of our choosing by use of the unitary $X_1$.

### $\bigoplus \mathbf{L} \subseteq \mathbf{BQ1P}$

The well known complexity class $\bigoplus \mathbf{L}$, (see *e.g.* [3],) may be defined as comprising those languages that can be decided by a uniform family of circuits composed entirely of CNot gates, whereby the $|x|$th circuit decides on the string $x$ by acting on the pure state $|\,x\,\rangle$ prior to a (deterministic) measurement of the first qubit in the computational basis. Equivalently, using our idea of "input as classical control", we could define $\bigoplus \mathbf{L}$ by having a uniform family of circuits composed entirely of classical-input-controlled CNot gates, in like fashion to the DQC1 circuit definition. We can use this idea to show :

**Lemma 2** *Class inclusion :* $\bigoplus \mathbf{L} \subseteq \mathbf{BQ1P}$.

Suppose we had a uniform circuit family, the circuits being made up exclusively of CNot gates, (the bits of an input string $x$ being used as an additional control over some or all of those CNot gates,) and suppose those circuits were intended to act on initial *pure* states of the form $|\,100...00\,\rangle$. Suppose we were then to measure a given bit of the output. We could describe the process by writing

$$C_{|x|}(x) \quad = \quad \Big|\, \langle\, 0\,|_1 \; Tr_{[2..w]}[\; U_{|x|}(x)|\,100...00\,\rangle\;]\;\Big|^2. \qquad (18)$$

Now suppose we were to make a dual family of circuits by taking every circuit from the original family and simply reversing the direction of each CNot gate therein, exchanging source qubit with target qubit, but retaining all of the classical controls on the gates from the bits in $x$. Write the action of this modified circuit as $\widehat{U}_{|x|}(x)$. We intend that circuits of this dual family $(\widehat{U}_n)_n$ be applied (with the same control from the input string $x$,) on the DQC1 state given in equation (2). It is easy to see that if an original circuit $U_{|x|}(x)$ would have mapped its pure starting state $|\,100...00\,\rangle$ to $|\,s_1 s_2 ... s_w\,\rangle$, say, where $s_1 = 1 - C_{|x|}(x)$, then the dual circuit $\widehat{U}_{|x|}(x)$ would map its mixed starting state $\rho_{\,\mathrm{start}}$ to the state $(1 + Z_1^{s_1} Z_2^{s_2} ... Z_w^{s_w})/2^w$. This is because whereas $\Lambda_i(X_j) \cdot |\,a\,\rangle_i |\,b\,\rangle_j = |\,a\,\rangle_i |\,a+b\,\rangle_j$, we have things the other way around in DQC1 notation :

$$\Lambda_i(X_j) \cdot Z_i^a \cdot Z_j^b \cdot \Lambda_i(X_j) \;=\; Z_i^{a+b} \cdot Z_j^b. \tag{19}$$

Consider next the somewhat larger DQC1 circuit composed as $\widehat{U}_n^\dagger \cdot X_1 \cdot \widehat{U}_n$. For a given control string $x$, this will map the state $\rho_{\,\mathrm{start}}$ to $\frac{1+\beta Z_1}{2^w}$, where $\beta = 2C_{|x|}(x) - 1$, as required, as the reader may readily check. Thus for such circuit families, we can guarantee determinism in the final measurement, and so certainly a **BQ1P** language is issued. This shows that $\bigoplus \mathbf{L} \subseteq \mathbf{BQ1P}$. ■

## More pure qubits

The next thing to show is that the complexity class **BQ1P** remains the same if we allow a constant number of pure qubits instead on just one, or even a logarithmic number.

To be precise, suppose we generalise the model by having $c$ pure qubits and being allowed, after arbitrary unitary processing, to measure just whether the first $d$ qubits are *all* in the state $|\,0\,\rangle$. For each circuit $U(x)$ we define $\beta_{c,d}$ accordingly, (*cf.* eqn (4)) :

$$\frac{1+\beta_{c,d}}{2} \;:=\; Tr\left[\; U(x) \cdot \frac{(1+Z)^{\otimes c}}{2^w} \cdot U(x)^\dagger \cdot \frac{(1+Z)^{\otimes d}}{2^d} \;\right]. \tag{20}$$

There would then be no point in having $d - c > O(\log |x|)$, since the signal strength would drop off superpolynomially for all $U(x)$, and so there would be no way to satisfy the requirement of equation (11) anywhere. If we were to have $c \sim d \sim O(\log n)$ then equation (20) can be expanded to be the scaled sum of the traces of polynomially many unitary operators, and thus, we shall show, the action of the machine could be simulated in the

**BQ1P** model. There are essentially two ways to complete the simulation in this case, either by parallel application of polynomially many simulators followed by some classical accounting at the end, or by randomly choosing one of those unitary operators to run the simulation on, to obtain a $\beta$ corresponding to the *average* trace. The first technique is not apparently applicable to **BQ1P** methodology, because as we have already noted, there is no real notion of parallel composition in DQC1. But the second option can be simulated in **BQ1P**, and though it may provide a polynomially weaker signal, it won't be so weak as to violate the guarantee required by equation (11).

Let us illustrate this simulation in more detail for the important case of $c = 2, d = 1$, (using the temporary notation "**BQ2P**" to indicate the class generated from allowing for two pure qubits instead of one) :

**Lemma 3** *Class equality :* **BQ2P = BQ1P**

The probability that we wish to sample is given by

$$\begin{aligned} \frac{1+\beta_{2,1}(x)}{2} &= Tr\left[ U(x) \cdot \frac{(1+Z_1)(1+Z_2)}{2^w} \cdot U(x)^\dagger \cdot \frac{(1+Z_1)}{2^1} \right] \\ &= \frac{1}{2} + \frac{\frac{Tr[UZ_1U^\dagger Z_1]}{2^w} + \frac{Tr[UZ_2U^\dagger Z_1]}{2^w} + \frac{Tr[UZ_1Z_2U^\dagger Z_1]}{2^w}}{2}, \end{aligned} \tag{21}$$

where $U$ (and hence $\beta_{2,1}$) depends on the input string, $x$. Now there is a DQC1 circuit of width $w + 2s$ that gets very close to sampling the probability $\frac{1+\beta_{2,1}/3}{2}$, namely

$$U(x) \cdot \Lambda_{1,w+2s}(X_2) \cdot \Lambda_{2,w+2s-1}(X_1) \cdots \Lambda_{1,w+2}(X_2) \cdot \Lambda_{2,w+1}(X_1). \tag{22}$$

This approximation works 'exponentially well' because the Toffoli gates that source the extra $2s$ qubits are sourcing totally random qubits, and so they reduce to random CNot gates, which establish a rapid-mixing Markov process for simulating the alternative starting signal; this being an almost-uniform mix from the set $\{Z_1, Z_2, Z_1Z_2\}$ instead of the usual $Z_1$. So if there were a '**BQ2P** language' according to the semantic guarantee

$$\left|\beta_{2,1}(x)\right| \geq \frac{1}{q(|x|)}, \tag{23}$$

then the same language would be in **BQ1P** according to the semantic guarantee

$$\left|\beta(x)\right| \geq \frac{1}{3q(|x|)} - \frac{1}{3 \cdot 4^{s-1}}, \tag{24}$$

as the reader may easily check by considering the effect of the random Toffoli gates of the circuit in equation (22) on the starting state. Recall that $q$ is polynomially bounded, so $s$ need only be logarithmic. ■

This illustration will readily generalise to show that class **BQ1P** is not extended by increasing the number of pure input bits to a logarithmic amount, (nor indeed by increasing the number of output bits by any amount, under the measurement assumptions we made.)

## Reduction

We are now in a position to show that :

**Lemma 4** *Class reduction :* **BQ1P** *is closed under reduction by logspace deterministic processes.*

This is the usual notion of reduction employed in complexity theory, [2]. It will in turn justify our earlier assumptions in section 2 relating to reduction concepts.

The idea is to use lemmas 2 and 3, so that if $\mathcal{L} \in$ **BQ1P** and $R$ is a function whose every bit is in $\bigoplus \mathbf{L}$, (which is a more general thing than merely saying that $R$ is logspace deterministic computable,) then we can prove the 'pre-reduction' language $\mathcal{L}' = \{\ x : R(x) \in \mathcal{L}\ \}$ to be in '**BQ2P**', and hence in **BQ1P**.

So proceed by supposing that we do have the use of two pure qubits rather than just one, and then divide the available qubits into two partitions with one pure qubit in each partition. The first partition will be used to undergo the transformations of the circuit being implemented, while the second partition will be used to compute the bits of $R$ as they are needed. Each time bit $R_i(x)$ is needed to choose between unitaries for applying to the first partition, use the techniques described for lemma 2 to map the second partition from state $\rho_{\,\mathrm{start}} = (1 + Z_1)/2^w$ to state $(1 + (-1)^{R_i(x)} Z_1)/2^w$. Then apply the unitary on the first partition *controlled on* the pure qubit in the second partition, before 'uncomputing' the computed control bit to restore the second partition to its original $\rho_{\,\mathrm{start}}$ value. (The partitions do not become entangled nor even classically correlated, because the control is always on a bit known to be pure and in the computational basis.) Apply lemma 3 to finish the proof. ■

The closure under logspace reduction, stability under small pertubations to the model, and the various completeness properties, give us some reassurance that the class has been defined correctly in a 'natural' way.

# 4 Computational complexity conjectures

We subscribe to the school of thought which assumes that each of **BPP** $\neq$ **BQP** and **L** $\neq$ **P** is true but too difficult to prove. Consistent with this belief, here are some other conjectures :

**Conjecture 1** $\mathbf{P} \not\subseteq \mathbf{BQ1P}$

**Conjecture 2** $\mathbf{BQ1P} \not\subseteq \mathbf{BPP}$

**Conjecture 3** $\mathbf{BQP} \not\subseteq \mathbf{BPP}^{\mathbf{BQ1P}}$ *: No hybrid machine consisting of classical processors and quantum-uncorrelated DQC1 processors can convincingly simulate the action of a quantum Turing machine within polynomial time, even allowing for use of adaptive techniques (classical correlations.)*

Clearly these conjectures are too strong to hope to prove affirmatively, given the belief expressed above. For example, a proof of conjecture 1 would immediately establish that $\bigoplus \mathbf{L}$ (and hence also **L**) is a proper subset of **P**, while a proof of conjecture 2 or 3 would confirm our belief that quantum computation is indeed superpolynomially more powerful than its classical counterpart. Conjecture 1 is important, since a 'straw poll'[5] reveals that opinion is divided as to whether it is even *likely*, let alone provable. A popular approach is to conjecture that **BQ1P** = **BQP**, and so one should make a serious attempt to look for evidence to resolve this discrepancy. Evidence for conjecture 2 can be found in section 2, since the problem of Trace Estimation appears to be classically hard. Conjecture 3 is a bit more speculative and intuitive, capturing our opinion that having access to many pure qubits in the *same* processor really is a valuable resource.

### Evidence by relativisation

Our main reason for finding conjectures 1 and 2 plausible has to do with relativisation, using oracles. The standard model for oracles fits into the DQC1 model only by representing a classical oracle as an (unknown) non-uniform class of permutation matrices which we may use as building blocks within the circuits in a circuit family, but which we may not modify nor examine any description of.[6] The idea is that the complexity of computation

[5] Ok, my methods aren't always especially scientific, and I'm not admitting in print just how many experts confessed a strong opinion on this matter.

[6] We assume that we are allowed to implement certain derivatives within the circuit, such as "controlled-$U$", a polynomial number of times if necessary.

represented by that oracle can be utilised 'atomically' within the circuitry, despite the fact that DQC1 doesn't allow for direct concatenation of computations, nor 'inputs and outputs' in the traditional sense, (as discussed in the first section.)

Let $U$ refer to such a ($2^w$-by-$2^w$) matrix. Then it is clear that while **P** circuitry relativised to $U$ can report back any entry of the matrix, the best that **BQ1P** circuitry can do (see section 2) is to estimate $\beta = Tr[V_U]/2^{w'}$, where $V_U$ is some polynomial-length unitary 'word' in the algebra $\mathcal{A}_{w'}$, some of whose 'letters' may be the unknown $U$. That there is a definite separation here can be shown inductively. Let $U$ be a unitary matrix with a $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ in the top left corner and let $U'$ be the same matrix but with the top left corner replaced by $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, *i.e.* $U' = U - 2|-\rangle\langle -|$. Whereas **P** circuitry is able to distinguish between these two oracles, we shall show that the difference $Tr[V_U] - Tr[V_{U'}]$ cannot be significant, *i.e.* cannot be on the order $2^{w'}/poly(w)$. First consider the case $w' = w$ and use induction on the 'word-length index' $t$ found in the expression $V_{U,t} = A_t \cdot U \cdot A_{t-1} \cdot U \cdots U \cdot A_1$. Now define

$$\chi_t \quad := \quad \max_W \; \big|\, Tr[\, W \cdot (V_{U,t} - V_{U',t})\, ] \,\big|, \quad W \text{ unitary.} \qquad (25)$$

From here we see

$$\begin{aligned} & \big|\, Tr[\, V_{U,t}\, ] \;-\; Tr[\, V_{U',t}\, ] \,\big| \qquad (26) \\ &\leq \; \chi_t \\ &= \; \big|\, Tr[\, W \cdot A_t \cdot (U \cdot V_{U,t-1} - U' \cdot V_{U',t-1})\, ] \,\big| \\ &= \; \big|\, Tr[\, (WA_tU)(V_{U,t-1} - V_{U',t-1})\, ] \;+\; 2\langle -|V_{U',t-1}WA_t|-\rangle \,\big| \\ &\leq \; \chi_{t-1} + 2, \end{aligned}$$

and so this difference between the traces cannot grow to the exponential required within polynomial word-length $t$. The argument is very similar for $w' > w$, so we shall leave details to the reader.

Therefore, if $\mathcal{O}$ is an oracle containing infinitely many (randomly chosen) examples of $(U, U')$ pairs, then (almost surely) one can construct deterministic distinguishing algorithms in **P** that have no analogue in **BQ1P**, and hence an oracle-based decision language in the former that cannot be contained in the latter, so $\mathbf{P}^{\mathcal{O}} \not\subseteq \mathbf{BQ1P}^{\mathcal{O}}$.

For conjecture 2, consider taking $U$ to be a random $2^w$-by-$2^w$ permutation matrix, implementing the permutation $\pi$ on $GF(2)^w$, subject to a certain promise (to be specified shortly,)

and let $V = H^{\otimes w} \cdot U \cdot H^{\otimes w} \cdot U \cdot H^{\otimes w} \cdot U$. Then

$$\begin{aligned} \frac{Tr[V]}{2^w} &= 2^{-5w/2} \sum_{ikm} (-1)^{i\cdot\pi(k) \,\oplus\, k\cdot\pi(m) \,\oplus\, m\cdot\pi(i)} \qquad (27) \\ &= 2^{w/2}\big(\, 1 - 2\ \mathbb{E}[\ i\cdot\pi(k) \ \oplus\ k\cdot\pi(m) \ \oplus\ m\cdot\pi(i)\ ]\ \big), \end{aligned}$$

and provided we have the promise that this value is sufficiently bounded away from 0, it is easy for **BQ1P** to resolve whether the value is closer to 1 or $-1$, but clearly any (randomised) classical device would have to query $U$ at exponentially many places to establish that distinction. Therefore any $\mathcal{O}$ containing infinitely many (randomly chosen) examples of such $U$ matrices will (almost surely) exemplify a case where $\mathbf{BQ1P}^{\mathcal{O}} \not\subseteq \mathbf{BPP}^{\mathcal{O}}$.

## Acknowledgements

Our thanks go to Richard Jozsa and Aram Harrow, for much proof-reading and encouragement. This work has been sponsored by CESG, (UK Government.)

## References

[1] M. Nielsen & I. Chuang, *Quantum Computation and Quantum Information.* (Cambridge University Press, 2000).

[2] C. H. Papadimitriou, *Computational Complexity.* (Addison-Wesley 1994).

[3] S. Aaronson & D. Gottesman, *Improved Simulation of Stabilizer Circuits* Physical Review A 70, 052328, 2004 (quant-ph/0406196).

[4] B. M. Terhal, D. P. DiVincenzo, *Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games* (quant-ph/0205133).

[5] E. Knill, R. LaFlamme, *Power of One Bit of Quantum Information* Phys. Rev. Lett. 81 (25), pp 5672 (1998).